涉密论文 □　公开论文 ☑

浙 江 大 學

# 本科生毕业论文


题目　知识驱动的物联网嵌入式固件自动化根因分析方法研究

姓名与学号　张乔 3200102817

指导教师　纪守领

年级与专业　2020级 计算机科学与技术

所在学院　计算机科学与技术学院

递交日期　递交日期

# 浙江大学本科生毕业论文（设计）承诺书

1. 本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2. 本人在毕业论文（设计）中除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得**浙江大学**或其他教育机构的学位或证书而使用过的材料。

3. 与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

4. 本人承诺在毕业论文（设计）工作过程中没有伪造数据等行为。

5. 若在本毕业论文（设计）中有侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

6. 本人完全了解**浙江大学**有权保留并向有关部门或机构送交本论文（设计）的复印件和磁盘，允许本论文（设计）被查阅和借阅。本人授权**浙江大学**可以将本论文（设计）的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编本论文（设计）。

作者签名：　　导师签名：

签字日期：　　年　月　日　　签字日期　　年　月　日

# 致 谢

# 摘 要

# Abstract

# 目录

# 第一部分

## 毕业论文

# 1 绪论

## 1.1 课题背景与意义

随着硬件技术和网络通信技术的快速发展，越来越多的设备和系统被连接到互联网，实现了智能化的生活和工作环境。嵌入式固件是物联网设备的核心，它使得这些设备能够实现数据采集、通信、控制等功能，从而为用户提供更便捷、智能的服务体验。同时物联网技术的应用已经渗透到各个产业领域，包括国防、工业、家居、医疗、交通等行业。嵌入式固件在这些领域中起着关键作用，它使得设备和系统能够实现自动化、智能化的控制和管理，提高生产效率、节约资源、降低成本。随着物联网设备的普及，信息安全和隐私保护越来越受到学术界与工业界的关注。嵌入式固件作为物联网设备的核心软件，需要具备良好的安全性和防护能力，以防止未经授权的访问、数据泄露、设备篡改等安全威胁，因此需要对其安全性进行检查。IBM Security 发布的《2022 年数据泄露成本报告》[1]表明，关键性基础设施在采用零信任安全策略（Zero Trust）方面还很滞后，其数据泄露的平均成本高达 540 万美元，比已采用零信任策略的组织高出 117 万美元。而关键基础设施中含有着众多的固件，这无疑说明固件安全是继续考虑的。HP Wolf security 发布 Threat Insights Report Q1 2022[2]表明，的随着劳动力的去中心化和混合办公模式的流行，端点安全的管理模式被彻底打破。设备不在现场供 IT 团队访问的混合办公环境中。更多的端点位于企业网络的保护之外也会降低可见性并增加通过不安全网络访问企业内部网络的攻击风险。购买远程办公设备的办公室工作人员中有 68% 表示安全不是他们采购决策的主要考虑因素。此外，43% 的人没有让 IT 或安全部门检查或安装他们的新笔记本电脑或 PC。如果不关注固件安全，这些多端的固件存在着严重的风险。

模糊测试技术可以通过将自动或半自动生成的测试输入到程序中，监测程序是否发生崩溃；而根因分析技术可以根据模糊测试的结果得到导致程序崩溃的测试输入，并寻找到程序中存在漏洞的位置。然而，目前尚未有针对运行资源受限的物联网嵌入式固件的自动化根因分析方法，无法针对嵌入式固件崩溃进

行深入分析。同时为了满足低级硬件中受约束的计算资源的需求，嵌入式固件通常被剥离调试信息，调试信息通常包含了变量名、函数名、文件名、行号等关键信息，有助于理解代码的结构和逻辑。当这些信息被剥离时，只留下一堆混合着数据的连续指令，这使得分析人员将失去对代码执行上下文的理解，更难理解代码的语义，从而导致手动根因分析变得更加困难。同时由于逆向执行时存在内存别名等问题，同时难以得到程序运行时的控制流与数据流，传统的自动化根因分析工具难以生效。

综上所述，针对物联网嵌入式固件设备，尚未存在有效的自动化根因分析方法衔接模糊测试，无法确定固件发生崩溃崩溃的根本原因并加以修正。因此，设计针对嵌入式固件的自动化根因分析工具是当前急需解决的问题。

## 1.2 国内外研究现状

### 1.2.1 嵌入式固件模糊测试技术

嵌入式固件运行在各种硬件平台上[3][4]，并与多个复杂的外围设备交互。早期的一些研究直接在真实硬件设备上进行黑盒 fuzzing，但这种方法缺乏全面的覆盖引导。例如，$\mu$AFL[5]和 SyzTrust[6]利用硬件调试器收集运行时信息，并支持嵌入式跟踪宏单元（ETM[7]）功能。其他研究采用硬件在环（hardware-in-the-loop）方法[8]将硬件请求转发到真实的外围设备。然而，这些方法需要频繁的上下文切换和模拟器与硬件之间的状态同步，导致显著的性能开销。

近期的工作采用重宿主技术以提高可扩展性，这种技术的核心思想是模拟外设并为固件提供有效输入，主要帮助固件通过初始化阶段的状态检查这些重宿主工作主要专注于使 IoT 固件能够进行 fuzzing，而不是专门优化后端 fuzzers（例如 AFL[9]）以增强 fuzzing 过程的效果。

### 1.2.2 自动化故障定位技术

虽然 fuzzing 有助于发现大量导致崩溃的测试用例，揭示潜在的漏洞，但识别这些崩溃的根本原因所需的最终手动调查是一个耗时且乏味的过程。这促使

了自动化故障定位技术的发展。

频谱基方法通过分析崩溃和非崩溃测试用例，根据它们出现的频率来对可疑指令进行排名[10][11][12]。虽然有效，但这些方法在分析粒度上存在限制。更先进的技术探索初始崩溃附近的不同执行路径，并为程序实体分配分数[13][14]。然而，分析每个个别测试用例的重时间成本使它们对于大规模固件 fuzzing 来说不切实际。

基于事后的方法利用崩溃后的遗留物，如执行轨迹和内存转储，反向分析程序并识别可能负责崩溃的最小指令集。例如，CrashLocator[15]利用崩溃报告中的崩溃栈信息定位故障函数，RETracer[16]反向分析代码和栈以找出错误值的传播方式。

此外，基于学习的方法也被广泛研究，以更好地理解崩溃。这些方法提出了神经网络架构来分析并将程序上下文纳入可疑分数计算。最近，随着大型语言模型（LLMs）的兴起，大模型辅助进行根因分析领域取得了积极的进展[17][18]

## 1.3 研究的主要难点

**运行时资源有限** 嵌入式固件通常部署在资源受限的设备上，这些设备缺乏足够的调试机制。理想情况下，根本原因分析方法需要对崩溃时的运行时信息有全面的了解，如执行轨迹、寄存器和内存值等。以往的研究通常依赖于核心转储来恢复这些运行时信息，但这严重依赖于崩溃时未损坏的栈数据和部分内存。与操作系统相比，嵌入式固件缺少用于检测崩溃（例如，sanitizers）和记录运行时信息（例如，核心转储）的机制。有时，开发者甚至需要采用原始的方法手动监控设备状态和通信，如 LED 闪烁和串行打印[19]。

**调试信息有限** 嵌入式固件通常被简化为原始二进制文件，以满足低级硬件中受限的计算资源需求。在此过程中，调试符号（如函数和变量名称）会被剥离，只留下一连串的指令和数据，这使得分析人员难以理解代码的语义。剥离后的固件二进制文件中缺乏语义信息，使得分析人员面对大量可能可疑的指令，而这些指令仍需进行繁琐的手动调查。然而，以往的工作要么将所有指令一视同仁，

要么设计上无法区分同一基本块内的不同指令，这对最终的手动根本原因调查提供了有限的实际指导。

## 1.4 研究内容与贡献

本文研究的主要内容是物联网嵌入式固件自动化根因分析方法。本文提出了一种针对物联网固件程序的框架，针对上节所提到的难点，本文进行了设计研究，主要贡献如下：

（1）设计了一种高效的运行时记录方法。针对嵌入式固件运行时资源有限以及调试信息有限的特点，我们设计了一种高效的运行时信息记录的方法。这种方法的核心思想是崩溃复现过程中，关注具体内存访问行为，并在故障定位中显著加速解决内存别名问题。具体内存访问包括指令的地址、访问方法（读或写）、源（目的地）寄存器，以及具体的目的地（源）内存地址。

（2）实现了 arm 平台下的自动化根因分析技术。我们设计了基于模糊测试结果的针对 arm 平台下的物联网嵌入式固件的根因分析方法。通过设计了一种高效运行时记录方法，构造使用-定义链，以及逆向执行等方法，分析模糊测试中的崩溃案例，分析导致崩溃的根本原因，定位固件漏洞

## 1.5 本文的组织结构

本文针对物联网嵌入式固件自动化根因分析方法这一问题，对现有自动化根因方法进行总结分析，提出了一种针对嵌入式固件的自动化分析方法，实现了对模糊测试中发生崩溃的固件漏洞检测。

本文共分为 5 章，其中各个章节具体内容安排如下：

第一章为绪论，介绍本文所研究的物联网嵌入式固件自动化根因分析方法这一主题的研究背景、意义与研究现状，并指出了研究难点与解决思路。

第二章为相关技术研究，介绍了主流自动化根因分析方法与物联网嵌入式固件自动化分析方法所面临的问题与解决方法。

第三章为本文主要工作，首先对物联网嵌入式固件自动根因分析方法框架

进行概述，然后并进行了问题定义与别名问题分析。最后从高效的运行时信息记录方法，使用定义链与逆向执行三个主要设计结构等三个方面对本文设计的方法进行介绍。

第四章为实验与评估，介绍本文所提出的方法所使用的实验环境，给出对于物联网嵌入式固件自动根因分析方法。

第五章为总结与展望，总结本文主要工作与贡献，提出本文工作中的局限性，并思考后续在此基础上可以继续发展与设计的工作

## 1.6 本章小结

本章中，我们首先指出了物联网嵌入式固件的重要性，分析了其自身存在的安全问题，论证了自动化分析方法研究的重要性。之后我们介绍了目前对于物联网嵌入式固件自动化根因分析方法研究的现状以及所面临的主要难点，讨论了当前主要研究方法的不足，思考了解决思路与后续可能的研究趋势，并介绍了本文的研究内容与贡献。最后我们介绍了本文的章节安排与组织结构。

# 2 自动根因分析相关技术研究

# 3 物联网嵌入式固件自动化根因分析

# 4 实验与评估

# 5 总结与展望

# 参考文献


[1] Cost of a Data Breach Report 2023[J].

[2] HP-Wolf-Security-Threat-Insights-Report-Q1-2022[J].

[3] CHEN J, DIAO W, ZHAO Q, et al. IoTFuzzer: Discovering Memory Corruptions in IoT Through App-Based Fuzzing[C]//Proceedings 2018 Network and Distributed System Security Symposium. Internet Society, 2018.

[4] LIU C, YAN X, FEI L. SOBER: Statistical Model-Based Bug Localization[J].

[5] LI W, SHI J, LI F, et al. $\mu$AFL: Non-Intrusive Feedback-Driven Fuzzing for Microcontroller Firmware[C]//Proceedings of the 44th International Conference on Software Engineering. 2022.

[6] WANG Q, CHANG B, JI S, et al. SyzTrust: State-Aware Fuzzing on Trusted OS Designed for IoT Devices[Z]. 2023.

[7] Embedded Trace Macrocell Architecture Specification[J]. 2011.

[8] KOSCHER K, KOHNO T, MOLNAR D. SURROGATES: Enabling Near-Real-Time Dynamic Analyses of Embedded Systems[J].

[9] ZALEWSKI M. American Fuzzy Lop[Z]. https://lcamtuf.coredump.cx/afl/. Apr. 2024.

[10] ABREU R, ZOETEWEIJ P. On the Accuracy of Spectrum-Based Fault Localization[J].

[11] ABREU R, ZOETEWEIJ P, VAN GEMUND A. An Evaluation of Similarity Coefficients for Software Fault Localization[C]//2006 12th Pacific Rim International Symposium on Dependable Computing (PRDC'06). IEEE, 2006.

[12] JONES J A, HARROLD M J. Empirical Evaluation of the Tarantula Automatic Fault-Localization Technique[C]//Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering. ACM, 2005.

[13] BLAZYTKO T, SCHLÖGEL M, ASCHERMANN C, et al. AURORA: Statistical Crash Analysis for Automated Root Cause Explanation[J].

[14] ARUMUGA NAINAR P, CHEN T, ROSIN J, et al. Statistical Debugging Using Compound Boolean Predicates[C]//Proceedings of the 2007 International Symposium on Software Testing and Analysis. ACM, 2007.

[15] WU R, ZHANG H, CHEUNG S C, et al. CrashLocator: Locating Crashing Faults Based on Crash Stacks[C]//Proceedings of the 2014 International Symposium on Software Testing and Analysis. ACM, 2014.

[16] CUI W, PEINADO M, CHA S K, et al. RETracer: Triaging Crashes by Reverse Execution from Partial Memory Dumps[C]//Proceedings of the 38th International Conference on Software Engineering. ACM, 2016.

[17] SHANG X, CHENG S, CHEN G, et al. How Far Have We Gone in Stripped Binary Code Understanding Using Large Language Models[Z]. 2024.

[18] YANG A Z H, LE GOUES C, MARTINS R, et al. Large Language Models for Test-Free Fault Localization[C]//Proceedings of the 46th IEEE/ACM International Conference on Software Engineering. ACM, 2024.

[19] MAKHSHARI A, MESBAH A. IoT Bugs and Development Challenges[C]//2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). IEEE, 2021.

# 作者简历

**姓名：**张乔　**性别：**男　**民族：**汉

**出生年月：**2002 年 7 月 **籍贯：**黑龙江省大庆市

2017.09-2020.07 大庆实验中学

2020.09-2024.07 浙江大学攻读学士学位

**获奖情况：**

**参加项目：**

**发表的学术论文：**

# 本科生毕业论文（设计）任务书

一、题目：

二、指导教师对毕业论文（设计）的进度安排及任务要求：

起讫日期 20　年　月　日　至　20　年　月　日

指导教师（签名）_____ 职称 _____

三、系或研究所审核意见：

负责人（签名）_____

年　月　日

# 本科生毕业论文（设计）考核

一、指导教师对毕业论文（设计）的评语：

指导教师（签名）_____

年　月　日

二、答辩小组对毕业论文（设计）的答辩评语及总评成绩：

| 成绩比例 | 文献综述（10%） | 开题报告（15%） | 外文翻译（5%） | 毕业论文质量及答辩（70%） | 总评成绩 |
|---|---|---|---|---|---|
| 分值 | | | | | |

负责人（签名）_____

年　月　日

# 第二部分

## 毕业论文开题报告